OPEN VIRTUAL REALITY TESTBED

Approaches Using Virtual Environments with Mosaic

Sandy Ressler
Manager Open Virtual Reality Testbed
National Institute of Standards and Technology

ABSTRACT

This paper describes several ways to use a computer generated environment with Mosaic. Two approaches are taken. First to allow the user to interact with an environment displayed as part of the Mosaic document, an in-line image. Second to create an independent process with which the user can have high bandwidth interaction, i.e. real time manipulations, which can drive Mosaic remotely.

In the first case an in-line picture of a space can be “walked-through” by allowing the user to select buttons which moves the point-of-view of the user forward, backward, left or right. Each user selection causes a network request to the Mosaic server which renders a new scene and/or loads a new HTML page. This method has the advantage of being a pure Mosaic application and issues of portability are minimized, however at the cost of performance.

Using the second method, creating an independent process, a user is allowed to interact with a graphical process running on the user’s workstation. At appropriate times in the interaction the user selects an object and a URL associated with the object is sent to Mosaic, using the remote control facility in the X version of Mosaic, and the Mosaic process goes to that URL. This method allows the user to have a great deal of interaction in real time with dynamic feedback. However issues of portability become more important, as the real time graphics process may be dependent on the particular platform.

These techniques demonstrate that the use and interaction with spatially oriented information spaces are practical. Both techniques have advantages and disadvantages which are discussed.

1.0 Background

Mosaic, the first Internet “killer application”, has brought the Internet into the popular culture. Users can begin to use simple point and click interfaces rather than a myriad of arcane commands and protocols. As another step towards a more natural interface we propose to use a first person point of view (POV) model which allows a user to interact with information in an intuitive manner.

Surrogate travel, has been around in a graphical form for over 15 years. The concept of first person POV has been used to convey a sense of place to a user interacting with a computer. The nature of the interaction has varied quite a bit depending on the context of the application. In the late seventies the “Aspen Movie Map” work at the Architecture Machine Lab (now known as the Media Lab) at MIT pioneered the concept of real time surrogate travel with photo realistic images [Donelson] [Lippman]. The movie map work was extended to the use of these environments as front ends to information [Bolt] [Herot].

Examples of first person POV interfaces come from many domains. Games such as 7th Guest, Myst, and Jurrasic Park all allows the player to travel in a first person POV manner. Information organizers such Packard Bell‘s Navigator and Apples eWorld interface to on-line services all use a first person POV metaphor to organize information and give users a sense of place and structure. [refs]

Virtual environments, 3D computer generated spaces can be used as the front end to information spaces. Such concepts are not new, work currently taking place a Xerox Parc [cone tree ref], and Silicon Graphics [FSN ref] exploits the use of such spaces as a front end to information. Historically much of this work derives from the concept of surrogate travel.

One effort in particular, the call for the creation of VRML (Virtual Reality Markup Language) has stimulated a number of efforts to link virtual environment to the World Wide Web (WWW). A number of interesting language proposals [Labyrinth] and demonstration systems [WebOOGL] can be found at the VRML [VRML] site.

Coupling of virtual environments with Mosaic and WWW is only the next logical extension to these spatial types of interfaces. There are however a number of different ways one can look at the spatial environments presented to the user, in particular a model based or image based environment.

2.0 Model Based vs. Image Based Virtual Environments

Issues concerning the virtual environment itself can be viewed in two fundamentally different ways. In the first the computer contains a 3D model of the environment, i.e. polygons, which must be rendered for display. This type of environment allows arbitrary travel through the environment at a cost of display quality and compute power. Alternatively one can create an environment based on images such as exemplified by the Aspen Movie Map work. The quality of the images and perceived environment can be very high, however one can not travel around in a completely arbitrary manner.

Apple Computer is about to introduce QuickTime VR [Apple2] a commercial product which allows users to quickly build image based virtual environments. The major advantage of this approach is that it sidesteps the difficult task of model building. Imaged environments can be easily constructed simply by pasting together photographs of an environment from several viewpoints and orientation.

Both the model and image based virtual environments can be used effectively as front ends to information bases. In the case of a model based environment one attaches attributes such as a URL to objects in the scene. In an image based environment one defines 2D areas as “hot” points which the user may select. Model based environments are computationally more expensive and more difficult to construct, however they offer unparalleled freedom of movement for the user. The decision to use one approach or the other clearly depends on the requirements of the application and available resources.

Integration of a virtual environment with Mosaic can take two forms the in-line graphics or external application approaches. Both approaches have advantages and disadvantages as discussed in the following sections.

3.0 In-line Graphics Surrogate Travel

Using this method of surrogate travel the user is presented with a graphic that visualizes a first person point of view (POV) in the middle of a Mosaic page. In addition to the graphic the page contains buttons which allow the user to control movement through the space. When a button is selected the user “moves” through the space and the graphics is updated.

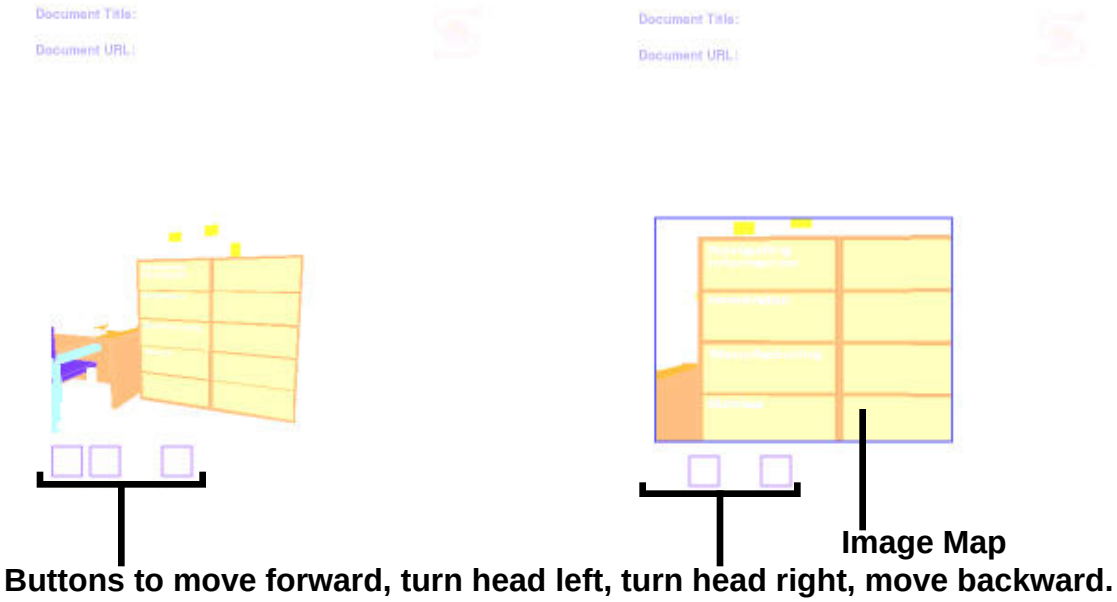

FIGURE 1. In-line Image Surrogate Travel and Mosaic

In the initial version of this experiment each scene was hand coded into a separate HTML page (clearly a hack). Subsequent to this a more extensive version allowing more complete travel through the room was developed by Allen Sun called the Virtual Corridor. In this version the users movement commands interact with a cgi-bin command which computes which image to use and automatically provides the linking.

4.0 External Graphics Surrogate Travel

Using this technique of surrogate travel, the user is presented with a completely separate application. This application “drives” mosaic via remote control. The principle advantage is that the user is able to manipulate a 3D environment in real time using all of the resources and performance of the graphics workstation, independent of the network bandwidth. The disadvantage is the loose coupling of the separate application with Mosaic and the need for a separate user interface.

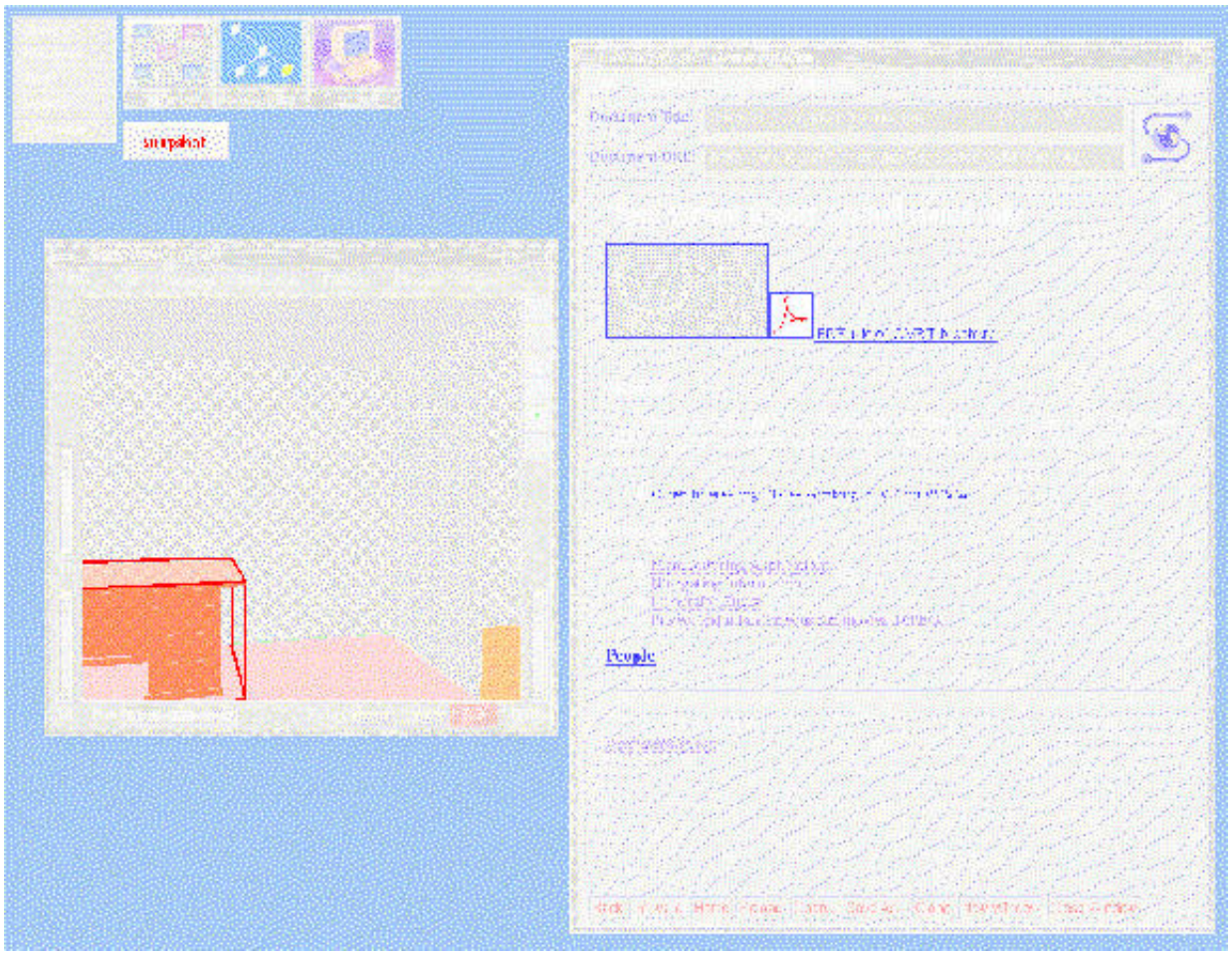

FIGURE 2. Workstation screen with URLViewer driving Mosaic

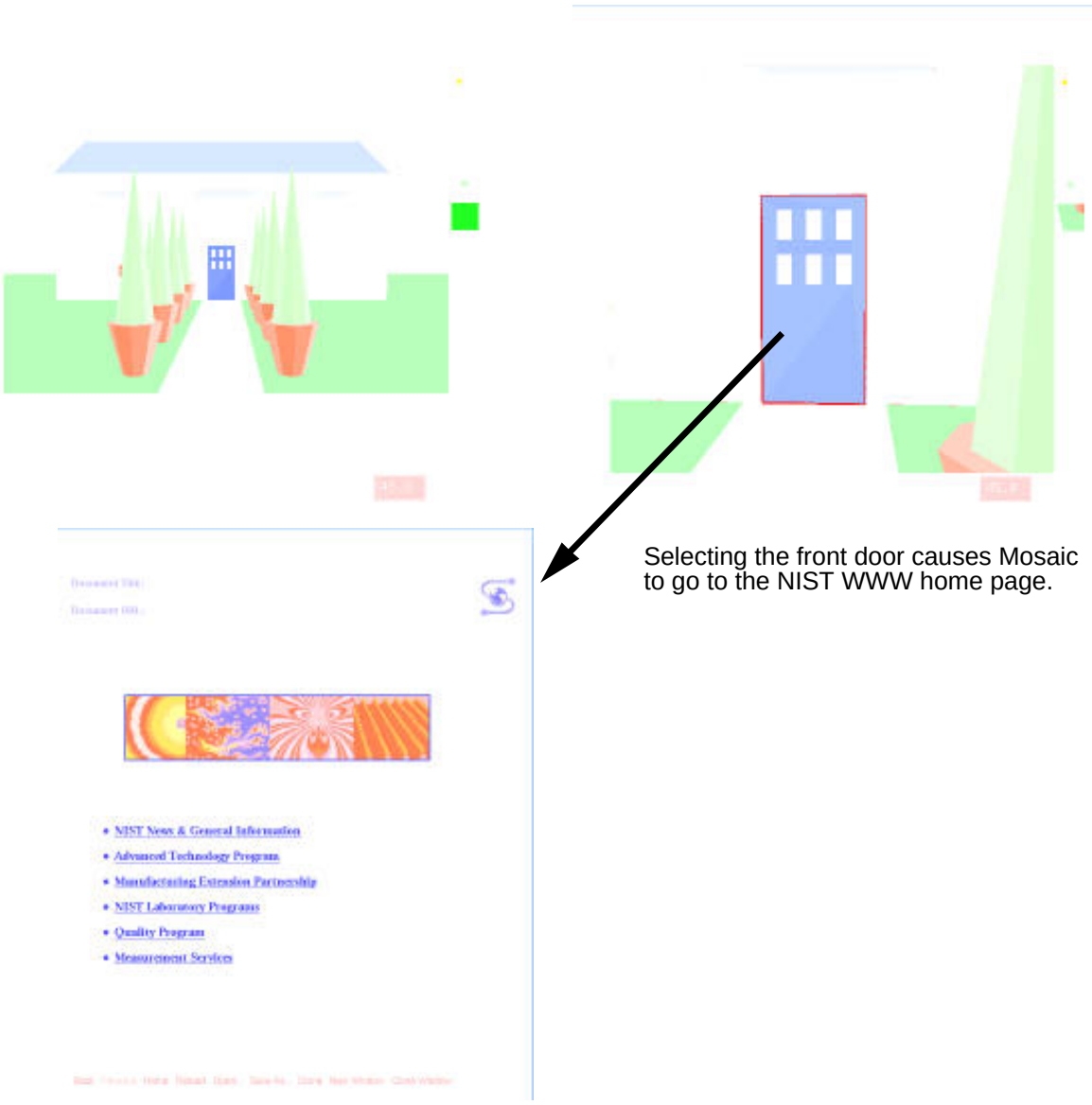

FIGURE 3. External application (URLScene) controlling Mosaic

The existing test application is a modification of the SceneViewer, Inventor application provided by Silicon Graphics [Inventor]. The SceneViewer program was modified to interpret object labels which are URLs. The URL is sent to a script which generate the appropriate SIGNAL causing Mosaic to go to the URL in the label. Following is a fragment of the Inventor file for the house containing a URL:

```
Separator {
Label {
label “http://nemo.ncsl.nist.gov/~sressler/OVRThome.html”
}
ShapeHints {
hints (SOLID | ORDERED | CONVEX)
creaseAngle 0.6
}
```

This method of control requires the ability of Mosaic to respond to a SIGNAL generated from an external process. NCSA Mosaic for the X Window System provides this feature and is documented at NCSA [Remote].

5.0 Future Directions

There are a number of directions these experiments can take. First it would be useful to extend the in-line graphic control to interact with a program that computes the views on the fly and return a new in-line image file. This close coupling of Mosaic with a 3D graphic program. This approach should also be practical for wide Mosaic distribution as the 3D computations and rendering takes place on the server and image files could be sent back to less powerful CPUs, i.e. PCs.

In addition to a closer coupling of the virtual environment generator an extension to the external application, URLviewer, to allow use of an immersive environment should prove useful. The addition of a 3D position tracker and head mounted display (HMD) would enable the user to become immersed in the environment and therefore indirectly into the data. Current HMDs however are not of sufficient quality to allow for any significant amount of textual reading.

Finally there is also an opportunity for the generation of VR clip art that could be used in conjunction with these approaches. 3D clip art has been around for a number of years [Swivel] [NEC]. These clip art packages provide the end user with manipulable images that can be positioned and rotated to any orientation and then included into documents. For the WWW one would like to travel to a location (i.e. a dinosaur museum) and get a 3D clip art icon meaningful to the user and place that object on ones virtual bookshelf. You then attach the URL to the object and now you have a 3D hotlist meaningful to the user.

6.0 Acknowledgments

Most of the actual labor to realize these experiments are due to a number of students I was fortunate to work with. Mark Pflaging a student at Georgetown University working for Joe Psotka at the Army Research Laboratory (the initial VR sponsor) modified the SceneViewer application. Allen Sun a student at MIT working at NIST glued the Inventor application to Mosaic and developed the Virtual Corridor in-line graphics program.

Sharon Davison a student a Catholic University and Mark Pflaging modeled the house for Sharon's dissertation.

7.0 References

[VRML] VRML information can be found at `http://www.wired.com/vrml/`

[Labyrinth] Labyrinth-VRML Specification by Parici and Pesce can be found at `http://www.wired.com/vrml/proposals/labspec.html`.

[WebOOGL] OOGL Proposed as Geometry Format for VRML by Munzer, Chi and Burchard of the Geometry Center in Minnesota can be found at `http://www.-wired.com/vrml/proposals/oogl.html`, and the WebOOGL Experiment at `http://www.geom/umn.edu/docs/weboogl/weboogl.html`

[Donelson] Donelson, W., "Spatial Management of Information", Proc. ACM SIGGRAPH 1978, Atlanta, Georgia.

[Lippman] Lippman, A. "Movie-Maps: An Application of the Optical Videodisc to Computer Graphics, Proc. ACM SIGGRAPH 1980.

[Bolt] Bolt, R., Spatial Data Management, DARPA Report Architecture Machine Group, MIT Cambridge, MA 1979.

[Herot] Herot, Carling, Friedell, Kramlich, "A Prototype Spatial Data Management System", Proc. ACM Siggraph 1980, Seattle WA.

[Inventor] SGI ref to product

[Remote] Using Mosaic by Remote Control is at `http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Docs/remote-control.html`

[Swivel] SwivelArt a 3D clip art collection from Macromedia, San Francisco CA.

[NEC] NEC Clip Art 3-D from NEC Technologies Inc.

"Extending WWW to support Platform Independent Virtual Reality" by Dave Raggett is at - `http://www.wired.com/vrml/concepts/raggett.html`

[refs for 7th Guest, Myst, Jurrasic park, Packard Bell, Apple eWorld

"Cyberspace" by Pesce, Kennard and Parisi is at - `http://www.wired.com/vrml/pesce/-www.html`

Mosaic and VR form first WWW conf.

[Apple2] Apple QuickTime VR

Mosaic Remote Control is described at - `http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Docs/remote-control.html`

Author Biography

Sandy Ressler ...

To contact the author email to sressler@nist.gov.